

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-117892

(43)Date of publication of application : 27.04.2001

(51)Int.Cl. G06F 15/173
G06F 15/177

(21)Application number : 2000-030406

(71)Applicant : NEC CORP

(22)Date of filing : 08.02.2000

(72)Inventor : TAKAGI HITOSHI

(30)Priority

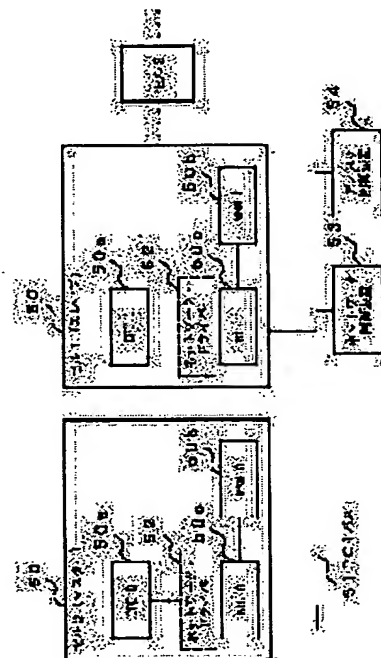
Priority number : 1999 417643 Priority date : 14.10.1999 Priority country : US

(54) MULTI-PROCESSOR COMPUTER SYSTEM FOR COMMUNICATION VIA INTERNAL BUS AND ITS COMMUNICATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an environment where processors execute communication via a connecting device through the used of a network protocol.

SOLUTION: A multi-processor computer system is provided with a plurality of computing cells having a processor, a memory and an interface unit and with the connecting device for connecting the plurality of computing cells through the use of the interface unit. The computing cells are respectively provided with a network driver for the connecting device and execute communication via the connecting device by using the network protocol.



LEGAL STATUS

[Date of request for examination] 16.01.2001

[Date of sending the examiner's decision of rejection] 19.11.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号 ✓
特開2001-117892
(P2001-117892A)

(43)公開日 平成13年4月27日(2001.4.27)

(51)Int.Cl.⁷

G 0 6 F 15/173
15/177

識別記号

6 7 6

F I

G 0 6 F 15/173
15/177

テーマコード(参考)

A 5 B 0 4 5
6 7 6 A

審査請求 有 請求項の数15 O L (全 13 頁)

(21)出願番号 特願2000-30406(P2000-30406)

(22)出願日 平成12年2月8日(2000.2.8)

(31)優先権主張番号 0 9 / 4 1 7 6 4 3

(32)優先日 平成11年10月14日(1999.10.14)

(33)優先権主張国 米国 (U S)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 ▲高▼木 均

東京都港区芝五丁目7番1号 日本電気株
式会社内

(74)代理人 100093595

弁理士 松本 正夫

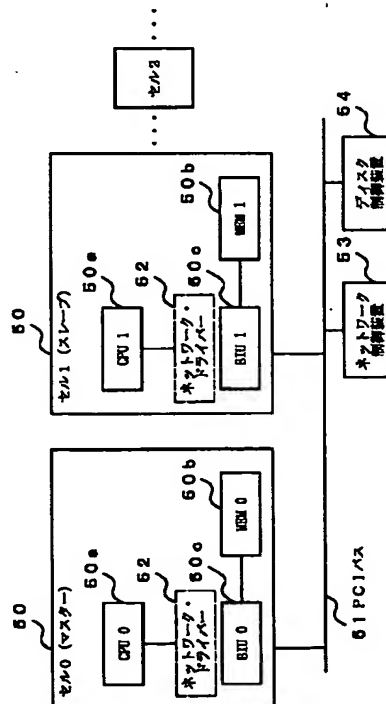
Fターム(参考) 5B045 AA03 BB12 BB17 BB28 BB29
BB42 BB47

(54)【発明の名称】 内部バスを介して通信するマルチプロセッサ・コンピュータシステムとその通信方法

(57)【要約】

【課題】 プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信する環境を提案する。

【解決手段】 マルチプロセッサ・コンピュータシステムであって、プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、コンピューティング・セルは、相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより相互接続装置を介して互いに通信する。



【特許請求の範囲】

【請求項 1】 マルチプロセッサ・コンピュータシステムであって、下記要件を備える。プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、

インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、

前記コンピューティング・セルは、前記相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【請求項 2】 前記相互接続装置が、業界標準の共用入出力バスであることを特徴とする請求項 1 に記載のマルチプロセッサ・コンピュータシステム。

【請求項 3】 前記共用入出力バスが、PCIバスであることを特徴とする請求項 2 に記載のマルチプロセッサ・コンピュータシステム。

【請求項 4】 前記相互接続装置が、交換相互接続装置であることを特徴とする請求項 1 に記載のマルチプロセッサ・コンピュータシステム。

【請求項 5】 前記ネットワーク・プロトコルが、TCP/IP プロトコルであることを特徴とする請求項 1 に記載のマルチプロセッサ・コンピュータシステム。

【請求項 6】 マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、下記ステップを備える。マルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、

相互接続装置を使用して前記コンピューティング・セルを接続し、

前記コンピューティング・セルに、相互接続装置用ネットワーク・ドライバを提供し、

前記相互接続装置用ネットワーク・ドライバにより、前記コンピューティング・セルがネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【請求項 7】 前記相互接続装置として、業界標準共用入出力バスを提供することを特徴とする請求項 6 に記載の通信方法。

【請求項 8】 前記共用入出力バスが、PCIバスであることを特徴とする請求項 7 に記載の通信方法。

【請求項 9】 前記相互接続装置が、交換相互接続装置であることを特徴とする請求項 6 に記載の通信方法。

【請求項 10】 前記ネットワーク・プロトコルが、TCP/IP プロトコルであることを特徴とする請求項 6 に記載の通信方法。

【請求項 11】 マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、下記

ステップを備える。少なくともマルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、

相互接続装置を使用して前記コンピューティング・セルを接続し、

前記コンピューティング・セルのプロセッサ間で前記相互接続装置を介してデータを送信するためにネットワーク・プロトコルを使用する。

10 【請求項 12】 前記相互接続装置として共用入出力バスを提供するステップをさらに備えることを特徴とする請求項 11 に記載の通信方法。

【請求項 13】 前記共用 I/O として PCIバスを提供するステップをさらに備えることを特徴とする請求項 12 に記載の通信方法。

【請求項 14】 前記相互接続装置として、交換相互接続装置を提供するステップをさらに備えることを特徴とする請求項 11 に記載の通信方法。

20 【請求項 15】 前記ネットワーク・プロトコルとして TCP/IP プロトコルを使用するステップをさらに備えることを特徴とする請求項 11 に記載の通信方法。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明は、マルチプロセッサ・コンピュータシステムの分野に関し、特に、ネットワーク・プロトコルを使用し、内部バスその他の相互接続装置を介してプロセッサが互いに通信するマルチプロセッサ・コンピュータシステムに関する。

【0002】

30 【従来の技術】ネットワーク・コンピューティング環境においては、ユーザー・アプリケーション・プログラムはネットワークのアプリケーション層プロトコルを使用して通信する。例えば、デスクトップ・システム（クライアント・マシン）上の電子メール・プログラムは、POP（Post Office Protocol）や SMTP（Simple Mail Transfer Protocol：メール転送プロトコル）などのメール転送プロトコルを使用してサーバ・コンピュータ上のメール・サーバ・プログラムと通信する。ネットワーク接続されたアプリケーション・システムの他の例では、ウェブ・ブラウザは HTTP（Hyper Text Transfer Protocol）を利用してウェブ・サーバと通信する。

【0003】アプリケーション・プログラムは、TCP/IP システムに取り付けた Sockets API、NetBIOS、TLI（Transport Layer Interface）API など、何らかの API（Application Program Interface：アプリケーション・プログラム・インターフェース）を介してネットワークと通信する。API の下

3

には、プログラム間でデータをやり取りする際に、データの喪失その他のエラーが発生することなく正常に転送されるようにする各種プロトコル・スタックの層がある。こうしたプロトコル・スタックの例としては、TCP/IP、SPX/IPX、NetBEUIなどが挙げられる。これらのプロトコルはそれぞれ、ネットワーク・ドライバと呼ばれるプログラムによって実装される。

【0004】ネットワーク・ドライバの下には、ネットワーク接続された2点間でデータ転送を行うために適切なハードウェアを実際に制御するハードウェア・ドライバがある。これらのハードウェア・ドライバとそれに対応するハードウェアは、イーサネット（IEEE 802.3）、ATM（Asynchronous Transfer Mode：非同期転送モード）、トークン・リング（IEEE 802.5）などの標準仕様や、ISDN（Integrated Subscriber Digital Network：統合ディジタル通信サービス網）、X.25、フレーム・リレー、DSL（Digital Subscriber Line）、ADSL（Asynchronous Digital Subscriber Lines）などのその他のWAN（Wide Area Networking：広域ネットワーク）標準を使用して実装される。最後に、コンピュータは、一般にこれらの標準の少なくとも1つに準拠するケーブルによって物理的に接続される。

【0005】一般に、ネットワーク・インターフェース・ハードウェアは、コンピュータの内部に設けられる。その例としては、イーサネット・カードなどのイーサネット・ハードウェア・コンポーネントをコンピュータのI/O（Input/Output：入出力）バスに接続した構成が考えられる。このようなイーサネット・カードは、Intel Corporationなどのベンダーによって製造されている。あるいは、2枚のイーサネット・カードをI/Oバスに取り付ける場合であれば、一方のカードをIntel製にし、もう一方のカードをCompaq Corporationなどの他のベンダーが製造したものとすることもできる。後者の場合、ネットワーク・ドライバの最下層が、コンピュータの構成とアプリケーション・プログラムから発行された要求を参照して、使用できるハードウェア・カードを特定する。

【0006】図10及び図11は、分散アプリケーション・システム内のネットワーク層の例である。図10は、Sockets、TLI、CPI/C、Named Pipes APIなどのトランスポート非依存API 11のサービスを起動するアプリケーション・プログラム10を示す。これらのAPIは、TCP/IP、SPX、NetBIOSを始めとする各種ネットワーク転送プロトコル12の実装へのアク

4

セスを提供する。転送プロトコル12の実装の下には、NDISやODIインターフェースなどの標準インターフェースによってアクセス可能な論理ネットワーク・ドライバ13の層がある。これらの論理ネットワーク・ドライバは、トークン・リング、イーサネット、ISDN仕様などの標準仕様を使って実装されたハードウェア14と協働して動作する。

【0007】図11は、最も一般的に使用されるネットワーク・ソフトウェアの層が、最上層にアプリケーション・プログラムのインターフェースを提供するアプリケーション層15を備えるOSI参照モデルの7つの論理層にどのようにマッピングされるかを示したものである。最下層の物理層21は、ネットワーク・ハードウェアと媒体に関する仕様を定義する。OSIモデルの中間層には、プレゼンテーション層16、セッション層17、トランスポート層18、ネットワーク層19及び論理リンク層20が含まれる。

【0008】インターネット・ウェブ・ブラウザ（Internet ExplorerやNetscape Navigatorなど）とインターネット・ウェブ・サーバとの通信の基礎をなす層について論じるため、その一例を図示する。図12に示すように、インターネット・ウェブ・ブラウザ22は、HTTPアプリケーション層プロトコルを使ってインターネット・ウェブ・サーバと通信する。ブラウザがHTTPによって通信するためには、TCP/IPプロトコルなどの下位レベルのプロトコルを使用して、すべてのデータがエラーやデータの喪失が発生することなく正常に送信されるようにする必要がある。

【0009】TCP/IP実装は、TCP/IPソフトウェアのサービスを起動するためのSockets APIと呼ばれるAPI 23を提供する。したがって、このAPIは、アプリケーション・プログラムがネットワーク・ドライバ24のサービスを起動するための一手段となる。ネットワーク・ドライバ・ソフトウェア24は、TCP/IPプロトコルを実装するために必要である。ネットワーク・ドライバ24は、ネットワークからの情報を転送、受信、または制御するためのread、write、及びioctlと呼ばれるライブラリー・ルーチンを呼び出すことによってハードウェア・ドライバ25を起動する。ネットワーク・ドライバ24には、オペレーティング・システムによって、コンピュータ内にイーサネット・アダプターまたはATMアダプターが備えられているかどうかといったネットワーク・ハードウェア26の構成情報が供給される。

【0010】そのため、マシン間のデータ転送における主要なオペレーションには、基本「送信」オペレーションが含まれる。したがって、この機能を起動するためのハードウェア・ドライバの基本オペレーションは、ネットワーク・ハードウェアにネットワークを介して供

給されたデータを送信させるためにハードウェア・ドライバから供給される SEND コマンド（または、関数呼び出し）である。同様に、READ の基本オペレーションでは、ハードウェア・ドライバがネットワークから送信されたデータを読み取る。

【0011】「Computer Architecture: A Quantitative approach（コンピュータ・アーキテクチャー： 定量的アプローチ）」、John Hennessy and David Patterson、第2版、Morgan Kaufman、1996 という題名のテキストには、従来のマルチプロセッサ・コンピュータシステムの例が示されている。このテキストに示される最初の例は、並行処理コンピュータシステムのノード群 31 を接続する汎用相互接続ネットワーク 30 を示す図 13 を参照して説明されている。例えば、MPP（Massively Parallel Processor）は、最大距離が非常に短い（25メートル未満のことも多い）何千ものノード群 31 を相互接続できる。

【0012】このテキストの2番目の例について説明する図 14 を参照すると、分散メモリ装置が示されている。この分散メモリ装置は、個別にプロセッサ、何らかのメモリ 34、典型的には何らかの I/O 35、及び相互接続装置 32 へのインターフェースを備えるノード群 33 で構成される。

【0013】これとは対照的に、数台のコンピュータシステムを相互に接続する従来のコンピュータ・ネットワークでは、コンピュータシステムはネットワーク・プロトコルを使用して、NIC（Network Interface Card）などのネットワーク・ハードウェアとネットワーク・ドライバを備えるネットワーク・ソフトウェアを介してネットワークと通信する。典型的には、コンピュータ・ネットワークのノード間の通信を調整するために、ネットワーク・オペレーティング・システム（NOS）が備えられる。

【0014】

【発明が解決しようとする課題】しかし、これまで開示された従来技術のシステムでは、内部または CPU バス、または類似の相互接続装置を別個のネットワークング装置として扱うことにより、マルチプロセッサ・コンピュータシステム内の異なるプロセッサがネットワーク・プロトコルを使用して互いに通信できるようにするものは存在しない。すなわち、上記の従来技術ではいずれも、複数のプロセッサが共用バスなどの適切な相互接続装置によって互いに接続されたマルチプロセッサ・コンピュータシステム内のプロセッサ間でネットワーク・プロトコルを使用する技術は開示されていない。したがって、上記の従来技術ではいずれも、マルチプロセッサ・コンピュータシステム内のプロセッサが標準ネットワーク・プロトコルを使用し、相互接続装置を介して互いに

通信できるようにするために、相互接続装置用ネットワーク・ドライバが提案されていない。

【0015】本発明の1つの目的は、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【0016】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【0017】本発明の他の目的は、プロセッサが TCP/IP プロトコルを使用し、共用 PCI バスを介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【0018】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【0019】本発明の他の目的は、プロセッサが TCP/IP プロトコルを使用し、交換相互接続装置を介して互いに通信するマルチプロセッサ・コンピュータシステムを提供することにある。

【0020】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【0021】本発明のさらなる目的は、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【0022】本発明の他の目的は、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【0023】本発明のさらなる目的は、プロセッサが TCP/IP ネットワーク・プロトコルを使用し、相互接続装置を介して通信するマルチプロセッサ・システムのプロセッサ間で通信する通信方法を提供することにある。

【0024】

【課題を解決するための手段】上記目的を達成するため、本発明によれば、マルチプロセッサ・コンピュータシステムであって、プロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルと、インターフェース・ユニットを使用して複数の前記コンピューティング・セルを相互接続する相互接続装置とを備え、前記コンピューティング・セルは、前記相互接続装置用のネットワーク・ドライバを備え、ネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【0025】請求項2の本発明によれば、前記相互接続装置が、業界標準の共用入出力バスであることを特徴とする。

【0026】請求項3の本発明によれば、前記共用入出力バスが、P C Iバスであることを特徴とする。

【0027】請求項4の本発明によれば、前記相互接続装置が、交換相互接続装置であることを特徴とする。

【0028】請求項5の本発明によれば、前記ネットワーク・プロトコルが、T C P / I Pプロトコルであることを特徴とする。

【0029】請求項6の本発明によれば、マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、マルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、相互接続装置を使用して前記コンピューティング・セルを接続し、前記コンピューティング・セルに、相互接続装置用ネットワーク・ドライバを提供し、前記相互接続装置用ネットワーク・ドライバにより、前記コンピューティング・セルがネットワーク・プロトコルを使用することにより前記相互接続装置を介して互いに通信する。

【0030】請求項11の本発明によれば、マルチプロセッサ・コンピュータシステムのプロセッサ間で通信する通信方法であって、少なくともマルチプロセッサ・コンピュータシステムのプロセッサ、メモリ、及びインターフェース・ユニットを備える複数のコンピューティング・セルを提供し、相互接続装置を使用して前記コンピューティング・セルを接続し、前記コンピューティング・セルのプロセッサ間で前記相互接続装置を介してデータを送信するためにネットワーク・プロトコルを使用する。

【0031】

【発明の実施の形態】本発明は、一般的な態様では、複数のコンピューティング・セルを備え、各セルがプロセッサ、メモリ、及びネットワーク・インターフェース・ユニットからなるマルチプロセッサ・コンピュータシステムを提供する。相互接続装置はコンピューティング・セルを接続する。コンピューティング・セルには、T C P / I Pネットワーク・プロトコルなどのネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信できるようにするための相互接続装置用ネットワーク・ドライバが設けられる。

【0032】他の一般的な態様では、本発明は、複数個のコンピューティング・セルを提供し、各コンピューティング・セルがプロセッサ、メモリ、及びネットワーク・インターフェース・ユニットを備えるマルチプロセッサ・システムのプロセッサ間で通信するための方法を提供する。接続ステップは、相互接続装置を提供することによってコンピューティング・セルを互いに接続する。

さらなる提供ステップは、各コンピューティング・セルに、T C P / I Pネットワーク・プロトコルなどのネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信できるようにするための相互接続装置用ネットワーク・ドライバを提供する。

【0033】したがって本発明は、内部バスが、メッセージがプロセッサ間で移動できるネットワーク・ハードウェアとして構成され定義される構成を提供する。ネットワーク・ドライバの最下層は、イーサネット・ハードウェア・カードなどのネットワーク・ハードウェアと類似した方法で、内部バスまたは類似のネットワーク装置としての相互接続装置を認識し通信できるようにプログラムされる。

【0034】ネットワーク・ドライバを内部バスまたは類似の相互接続装置と通信できるようにプログラムすることは、当該技術に精通した当業者が通常有する能力である。例えば、リナックス環境でネットワーク・ドライバをプログラムする方法の詳細は、インターネット上のURL “<http://www.linuxdoc.org/HOWTO/Ethernet-HOWTO-8.html#ss8.2>” で提供されている。

【0035】例えば、T C P / I P実装におけるこのようなネットワーク・ドライバの実装は、I Pパケットが形成されてから、ネットワーク・インターフェース・ハードウェアを介してネットワークに送信されるまでの間にそれを傍受するようにプログラムされたソフトウェアとすることができる。この特別プログラム式ソフトウェアは、I Pアドレスがマルチプロセッサ・システムの他のプロセッサの1つを指定しているかどうか判断し、指定している場合は、該当するプロセッサを宛先とし内部バスまたはその他の相互接続装置を介してパケットをルーティングする。I Pアドレスが他のプロセッサの1つに属しないと判断した場合、特別プログラム式ソフトウェアは、さらなる転送のため通常の方法で、ネットワークを宛先とし、ネットワーク・インターフェース・ハードウェアを介してI Pパケットをルーティングする。以下に、当該ソフトウェアがI Pパケットをプロセッサ間でルーティングするための手法の例を取り上げて詳細に説明する。ここでは、マルチプロセッサWindows NT環境の例として、タスク間メッセージングを使用して異なるプロセッサ間のデータ転送を行う場合を使用することができる。

【0036】図1は、本発明の好適な実施の形態によるシステム構成を示す図である。このシステムには数個のコンピューティング・セル50が備えられ、各セルはC P Uやメモリなどの従来のコンピュータ・コンポーネントで構成されている。各セルは業界標準I / OバスであるP C Iバス51に取り付けられている。各セルは、ネットワーク・プロトコルを使用し、共用バス51を介して互いに通信する。バス上で通信するためのネットワー

ク・プロトコルとしては、例えば、TCP/IPプロトコルを使用することができる。

【0037】図1に示すように、このマルチプロセッサ・システムは、PCIバス51に取り付けられた2以上のコンピューティング・セル50を備える。各セル50は、CPU 50a、メモリ装置(MEM) 50b、及びBIU (Bus Interface Unit) 50cを備える。セルの1つであるセル(0)は、マスター・セル50(マスター)と呼ばれ、その他のセルはスレーブ・セル50(スレーブ)と呼ばれる。各BIU 50cは、PCIバス51へのインターフェースを備える。マスター・セルのBIU 50c (BIU0)は、通常のPCIバス・インターフェースLSIである。このようなLSIの例としては、Intel Corporationの430TX LSIがある。このLSIの詳細は、Intel社のいくつかの刊行物に示されているほか、“<http://developer.intel.com/design/chipsets/mature/430tx/index.htm>”のアドレス(URL)のウェブ・サイトでも入手できる。

【0038】各セル50は、さらにCPUバス(PCIバスなど) 51をネットワーク・デバイスとして認識し、それと通信するプログラム式ネットワーク・ドライバ52も備えているので、TCP/IPネットワーク・プロトコルなどの標準ネットワーク・プロトコルを実装するネットワーキング・ソフトウェアは当該CPUバス51を通信媒体として使用できる。

【0039】スレーブ・セルのBIU 50c (BIU1など)も従来のPCIバスインターフェースLSIに類似しているが、以下のメモリ・エイリアス機能とDMA (Direct Memory Access: 直接メモリ・アクセス) 機能が追加されている点が異なる。CPUは、少なくとも32ビット・データ・バス及び32ビット・アドレス指定能力を有するのが望ましい。

【0040】PCIバス51には、さらに追加のハードウェアを取り付けることもできる。この追加のハードウェアとしては、例えば、マスター・セルがディスク・ドライブにアクセスするためのディスク制御装置54や、マスター・セルが他のシステムもしくはネットワークと通信するためのイーサネット制御装置などのネットワーク制御装置53が挙げられる。

【0041】本実施の形態においては、自セルのメモリは「ローカル・メモリ」として参照される。他セル内のメモリは、マスター・セル内のメモリを含み、「リモート・メモリ」として参照される。

【0042】図2は、サブセル150がCPU150aとキャッシュ155を備える代替の実施の形態を示す。各サブセル150はCPUバス156に接続されており、CPUバス156は共用メモリ装置157にも接続されている。共用メモリ装置157は、共用メモリ15

7bとメモリ制御装置157aを備えている。CPUバス156は、PCI制御装置151aを介してPCIバス151と相互接続されている。サブセル150と共用メモリ157のこの構成(図2内の点線部分)は、図1に示す実施の形態のセル50に対応する。したがって、サブセル150のこれらのグループは、図1のセル50がPCIバス51を使用して互いに接続されているのと同様に、PCIバス151を使用して互いに接続できる。同様に、PCIバス151は、例えばネットワーク制御装置153及びディスク制御装置154などを介して外部装置と接続できる。この構成では、プログラム式ネットワーク・ドライバ・ソフトウェア152は、プロセッサ150aがネットワーク・プロトコルを使用しCPUバス156を介して同じグループのサブセル内にある他のプロセッサと通信できるようにプログラムすることが可能である。さらに、プロセッサ150aは、サブセル150の他のグループに属するプロセッサ150aと通信することもできる。

【0043】上述したように、この特別プログラム式ネットワーク・ドライバは、当業者であれば通常有する能力の範囲内で数通りにプログラムできる。こうした実施例では、このプログラム式ドライバ・ソフトウェアをTCP/IP実装内でIPパケットを傍受するように実施して、他のプロセッサの1つを宛先としてアドレス指定されたIPパケットが、ネットワーク・インターフェース・ハードウェアを介してネットワークに送出されるのではなく、共用CPUバス156及び/またはPCIバス151上を送信されるようにすることが可能である。

【0044】図3～図8は、共用バスか類似の相互接続装置によって接続された、図1で開示されるセル間のデータ転送の例を示す概略図である。この概略図のうち、図3はマスターBIUによって実行される機能を示すフローチャートである。マスターBIUは、自セルCPUとPCIバスの両方から送られてくる要求を管理する。

【0045】ステップ605では、マスターBIUは、受信した要求がCPU要求かどうか判断し、CPU要求の場合は、ステップ610に進む。ステップ610では、マスターBIUはそのCPU要求のアドレスが最大設置メモリ50bより小さいかどうか判断し、小さい場合は、ステップ615でその要求の行き先はマスター・セルのメモリとされる。一方、CPU要求のアドレスがステップ610の最大設置メモリより大きい場合は、ステップ620でその要求の行き先はPCIバスとなる。

【0046】ステップ605で要求がCPU要求でないと判断した場合、BIUはステップ625に進み、その要求がPCI要求かどうか判断する。PCI要求と判断した場合、BIUはステップ630に進み、PCI要求のアドレスが最大設置メモリより小さいかどうか判断する。小さい場合は、ステップ640でその要求の行き先はマスター・セルのメモリ50bとなる。そうでない場

10

20

30

40

50

合、すなわちCPU要求のアドレスが最大設置メモリより大きい場合、ステップ635でその要求の行き先はマスター・セルの内部レジスタとなる。

【0047】図4は、スレーブBIUによって実行される機能を示すフローチャートである。スレーブBIUは、スレーブCPU、BIU内のDMAモジュール、及びPCIバスから送られてくる要求を管理する。ステップ705で、BIUは要求がCPU要求かどうか判断する。CPU要求と判断した場合、BIUはステップ710で、要求されたアドレスがスレーブ・セル内の最大設置メモリより小さいかどうか判断する。要求のアドレスが最大設置メモリより小さい場合は、ステップ715でその要求がスレーブ・セル内のローカル・メモリに向けられ、そうでない場合、ステップ720でその要求の行き先はPCIバスとなる。

【0048】ステップ705で要求がCPU要求でないと判断した場合、BIUはステップ725に進み、その要求がPCI要求かどうか判断する。PCI要求の場合、BIUはステップ730に進み、PCI要求のアドレスが最大設置メモリより小さいかどうか判断する。小さい場合は、ステップ735でその要求の行き先はローカル・メモリとなる。そうでない場合、すなわちPCI要求のアドレスが最大設置メモリより大きい場合、ステップ740でその要求の行き先はスレーブ・セルの内部レジスタとなる。

【0049】図5は、本発明のマルチプロセッサ・システム内で実行されるメモリ・エイリアス機能のステップを示すフローチャートである。前述したように、CPU要求の行き先はローカル・メモリかPCIバスのいずれかとなる。要求のアドレスが設置メモリの最高位アドレスより低い場合、その要求の行き先はローカル・メモリにされる。要求のアドレスが設置メモリの最高位アドレスより高い場合、その要求の行き先はPCIバスとなる。PCIバスのアドレス・スペースの最下部は、マスター・セル（図1ではマスターセル50（セル0））のメモリ・スペースに割り当てられる。

【0050】システム内の他のスレーブ・セルに備えられるメモリにアクセスできるようにするため、「エイリアス」手法も採用されている。「エイリアス」は、PCIバスのアドレス・スペースにアクセスすることによって各セルが他のセルのメモリにアクセスできるようにする機能である。PCIバス内の事前定義されたアドレスが、セル内のメモリにマッピングされる。各セルは、PCIバスのアドレス・スペース内に異なる基底アドレスを有する。例えば、図1を参照すると、PCIアドレス・スペース内のアドレス「0xa0000000」（0xは16進数を示す）は、セル0のメモリのアドレス「0x00000000」にマッピングされ、PCIバスのアドレス・スペース内のアドレス「0xa1000000」は、セル1のメモリのアドレス「0x0000

0000」にマッピングされ、PCIバスのアドレス・スペース内のアドレス「0xa2000000」は、セル2のメモリのアドレス「0x00000000」にマッピングされる。

【0051】各BIUは、最高位アドレスと最低位アドレスを指定する2個のレジスタを備える。これらのアドレスは、PCIバスからBIUの対応するセルのメモリに引き渡すことができる。これらのレジスタは、例えば、HIGH及びLOWレジスタのように命名される。各BIUは、PCIバス上で行われるバス・トランザクションを監視する。ここで図5を参照する。ステップ805及び810に示されるように、BIUはHIGH及びLOWレジスタによって指定される範囲内のアドレスに対するアクセス要求を検出すると、そのトランザクションに対して肯定応答を発行し、ステップ815で自セルのローカル・メモリにトランザクションを転送する。各セルのBIU内のHIGH及びLOWレジスタによって指定されるアドレス範囲は重複してはならない。HIGH及びLOWレジスタ内に格納される値は、ブートアップ時にシステムによって初期化される。

【0052】本発明は、さらに、各スレーブ・セル内にDMAモジュールを提供する。このDMAモジュールは、マスター・セルのメモリ（リモート・メモリ）とスレーブ・セルのメモリ（ローカル・メモリ）間でやり取りされるデータ転送要求を管理する。この転送を制御するための状態（すなわち、レジスタ）には数種類がある。これらのレジスタは、メモリ・マッピングによってアクセスされる。例えば以下のようなレジスタを備えることができる。

【0053】STATUSレジスタ： この32ビット・レジスタは、DMAが処理している転送の状況及びエラー情報を提供する。

【0054】SRCレジスタ： この32ビット・レジスタは、マスター・メモリ・ブロックとの間で送受信されるローカル・アドレス・ブロックの開始アドレスを保持する。

【0055】DSTレジスタ： この32ビット・レジスタは、転送中のローカル・メモリ・ブロックで上書きされるリモート・アドレス・ブロックの開始アドレスを保持する。

【0056】LENレジスタ： この32ビット・レジスタは、読み取り動作と書き込み動作によって振る舞いが異なる。このレジスタに対する書き込み動作では、SRC及びDSTを使用した転送が開始され、その後、書き込まれたデータが転送長を指定する。このレジスタからの読み取り動作では、現在実行中のデータ転送の残余長が捕捉される。

【0057】これらのレジスタは、事前決定されたPCIバスのアドレス・スペースにメモリ・マッピングされる。すなわち、各セルはPCIバス上の該当するアドレ

スから読み取ったり、それに書き込んだりすることによってこれらのレジスタにアクセスできる。

【0058】次に、本発明のシステムにおけるデータ転送動作について説明する。先に述べたように、ネットワーク通信におけるセル間の最も基本的な動作は送信動作である。この送信動作は、データ転送における発信セルと受信セルの組み合わせによって、(i) マスターからスレーブの転送、(ii) スレーブからマスターの転送、及び(iii) スレーブ間の転送、の3種類に分類される。次に、これら3種類の送信動作について個別に説明する。

【0059】マスターからスレーブへの転送

図6のフローチャートに示すように、マスター・メモリ内のデータ・ブロックが転送先スレーブBUI内のDMAモジュールによって転送されると、以下のステップが続いて実行される。すなわち、ステップ905で、マスターCPUが、未完了のトランザクションがあるかどうか判断するためにターゲット・スレーブ・セル内のSTATUSレジスタを読み取る。ステップ910で未完了のトランザクションがあると判断された場合、ステップ915で現在のトランザクションが事前に決定された時間にわたってスリープ状態に入り、これが経過するとステップ905及び910が繰り返される。一方、ステップ910で得られたSTATUSレジスタからの応答データが、ターゲット・セル内のDMAが新たな転送を開始できる状態にあることを示している場合は、ステップ920でマスターCPUが該当するアドレスを有するターゲット・セル内のSRCレジスタに書き込む。その後、ステップ925でマスターCPUはSRCレジスタへの書き込みが成功したかどうかを確認する。成功した場合、マスターCPUはステップ930に進み、ターゲット・セル内のLENレジスタに書き込みを行って転送を開始させる。

【0060】DSTレジスタは初期化時にスレーブCPUによって設定されるため、この時点ではDSTレジスタへの書き込みは行われない。STATUSレジスタによってDMAモジュールが新たな転送を開始できる状態にあることが示されている場合に、メモリ内の転送先データ・バッファが準備完了状態にあるかどうかを確認することはスレーブの責任となる。

【0061】転送実行中、ステップ935で、DMAモジュールは転送の進行にともなってLENレジスタの値を一定間隔で減少させる。LENレジスタが0となったら、DMAはステップ940でスレーブCPUへの割り込みを生成する。この割り込みによって、転送が終了し、新たなデータ転送に備えてDMAモジュールが再初期化される。

【0062】スレーブからマスターへの転送

図7のフローチャートに示すように、スレーブ・メモリ内のデータ・ブロックがマスターのCPU(CPU0)

に転送されるデータ転送では、以下のステップが実行される。この転送は、ステップ1005でスレーブCPUがマスターCPUに対する割り込みを生成することによって開始される。割り込みを受信すると、マスターCPUはステップ1010でPCIバス内のメモリ・エイリアス領域にアクセスすることによって、スレーブのメモリ・ブロックからデータを読み取る。その後、マスターCPUはステップ1015でデータを自セルのメモリに書き込み、スレーブ・セルからマスター・セルへのデータ転送を完了させる。

【0063】スレーブ間の転送

図8のフローチャートに示すように、転送元スレーブのメモリ内のデータ・ブロックが転送先スレーブのCPUに転送される場合には以下のステップが実行される。ステップ1105で、転送元スレーブのCPUは転送先スレーブのCPUに対する割り込みを生成する。ステップ1110で、転送先スレーブのCPUは、PCIバス内のメモリ・エイリアス領域にアクセスすることによって、転送元スレーブのメモリ・ブロックからデータを読み取る。その後、転送先スレーブのCPUはステップ1115でデータを自セルのローカル・メモリに書き込み、スレーブ間データ転送を完了させる。

【0064】本発明の第2の実施の形態では、共用バスではなく、2点間接続を使用するセル間接続方式が使用される。したがって、セル150は、例えば図9に示すような交換ファブリック161によって接続することができる。この交換ファブリック161の例としては、クロスバー・ネットワーク、多段交換ネットワーク、時分割共用バスなどが挙げられる。

【0065】この接続方式のデータ転送トランザクションは、例えば、転送元アドレス(セル: メモリ)、転送先アドレス(セル: メモリ)、転送タイプ、及び実際のデータ、というデータ・アイテムによって定義できる。

【0066】例えば、セル0～セル3のメッセージを送信する場合であれば、交換ファブリックを介して移動するトランザクションは以下のデータ値によって実行できる。

【0067】

40 転送元アドレス: ;セル0: 0x00a00000
転送先アドレス: ;セル3: 0x00040000
タイプ: ;書き込み
データ:

【0068】場合によっては、このデータ転送は共用バス通信方式を使用する実施の形態よりも簡単に実行できることがある。

【0069】以上好ましい実施の形態及び実施例をあげて本発明を説明したが、本発明は必ずしも上記実施の形態及び実施例に限定されるものではなく、その技術的思想の範囲内において様々に変形して実施することができ

る。

【0070】明細書に組み込まれその一部を構成する添付図面は、本発明の現在好適な実施の形態を図示し、上記の概略説明及び好適な実施の形態の詳細説明とともに、本発明の原理を説明するものである。

【0071】

【発明の効果】以上説明したように本発明によれば、プロセッサがネットワーク・プロトコルを使用し、相互接続装置を介して互いに通信する環境が実現される。

【0072】また、プロセッサがネットワーク・プロトコルを使用し、共用バスを介して互いに通信することができる環境が実現される。

【0073】例えば、プロセッサがTCP/IPプロトコルを使用し、共用PCIバスを介して互いに通信することが可能となる。

【0074】さらに、プロセッサがネットワーク・プロトコルを使用し、交換相互接続装置を介して互いに通信することができる環境が実現される。

【図面の簡単な説明】

【図1】 本発明の好適な実施の形態によるシステム構成を示すブロック図である。

【図2】 本発明の代替の実施の形態によるシステム構成を示すブロック図である。

【図3】 マスター・セルのBIU (Bus Interface Unit) によって実行される機能論理を示すフローチャートである。

【図4】 スレーブ・セルのBIUによって実行される機能論理を示すフローチャートである。

【図5】 メモリ・エイリアス機能のステップを示すフローチャートである。

【図6】 マスター・セルからスレーブ・セルへのデータ転送におけるステップを示すフローチャートである。

【図7】 スレーブ・セルからマスター・セルへのデータ転送におけるステップを示すフローチャートである。

【図8】 スレーブ・セル間のデータ転送におけるステップを示すフローチャートである。

【図9】 本発明の他の好適な実施の形態によるシステム構成を示すブロック図である。

【図10】 ネットワーク・プロトコル・スタックへのインターフェースを示す図である。

【図11】 ネットワーク・プロトコルの機能層を説明するOSI参照モデルを示す図である。

【図12】 典型的なインターネット・ブラウザ・アプリケーションのネットワーク・インターフェースを示す図である。

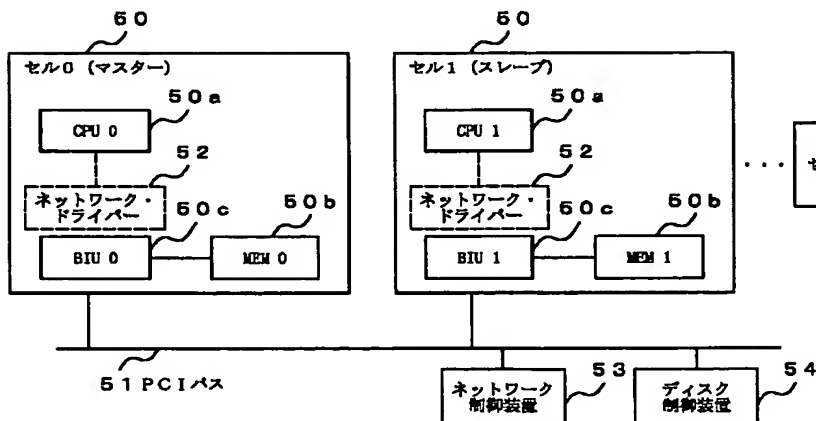
【図13】 汎用相互接続装置ネットワークを示す図である。

【図14】 分散メモリ・マルチプロセッサ・システムのアーキテクチャーを示す図である。

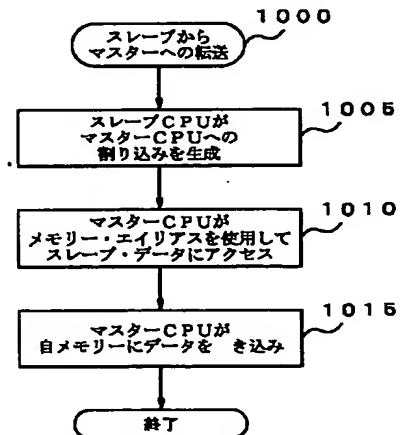
【符号の説明】

- 50 セル
- 50a CPU
- 51 PCIバス
- 52 ネットワーク・ドライバ
- 53 ネットワーク制御装置
- 54 ディスク制御装置
- 151a PCI制御装置
- 151 PCIバス
- 153 ネットワーク制御装置
- 154 ディスク制御装置
- 155 キャッシュ
- 156 CPUバス
- 157a メモリ制御装置
- 161 交換ファブリック

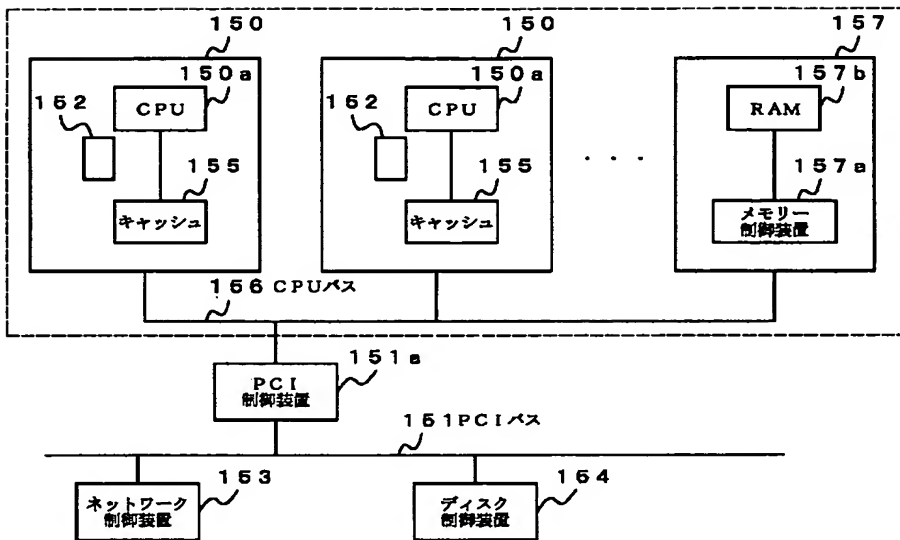
【図1】



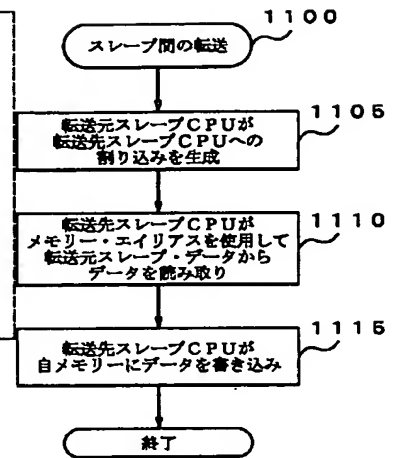
【図7】



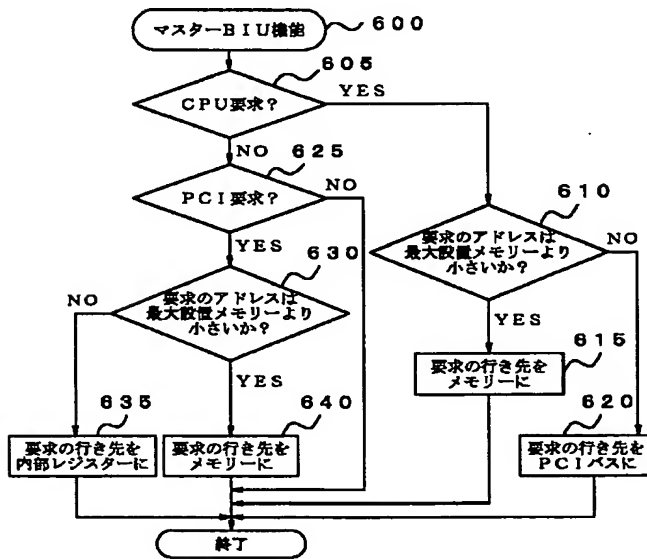
【図2】



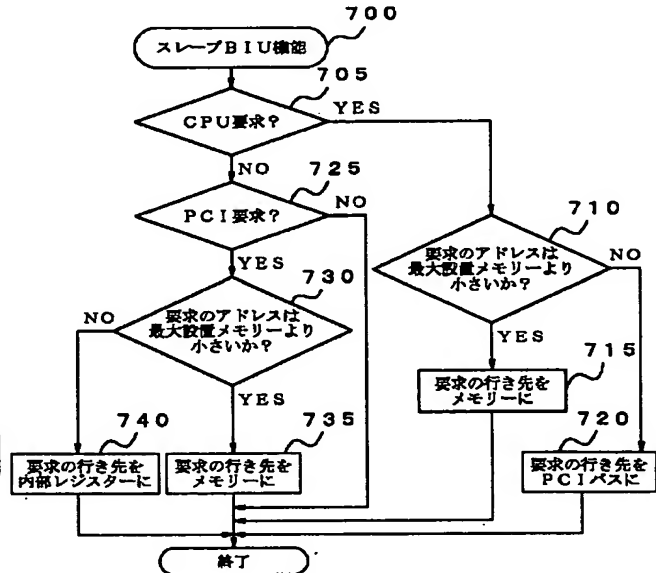
【図8】



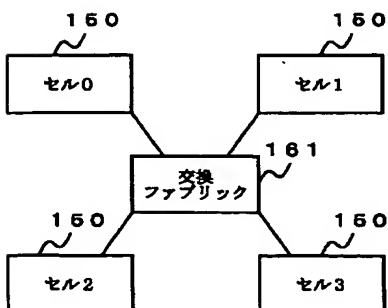
【図3】



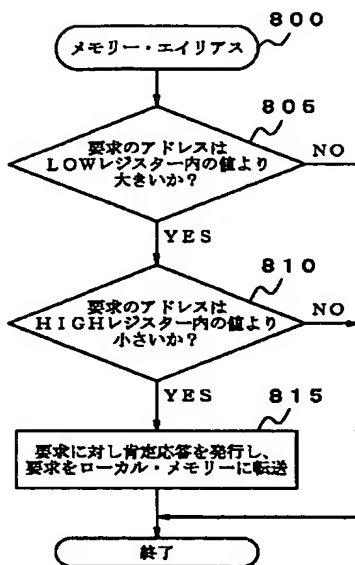
【図4】



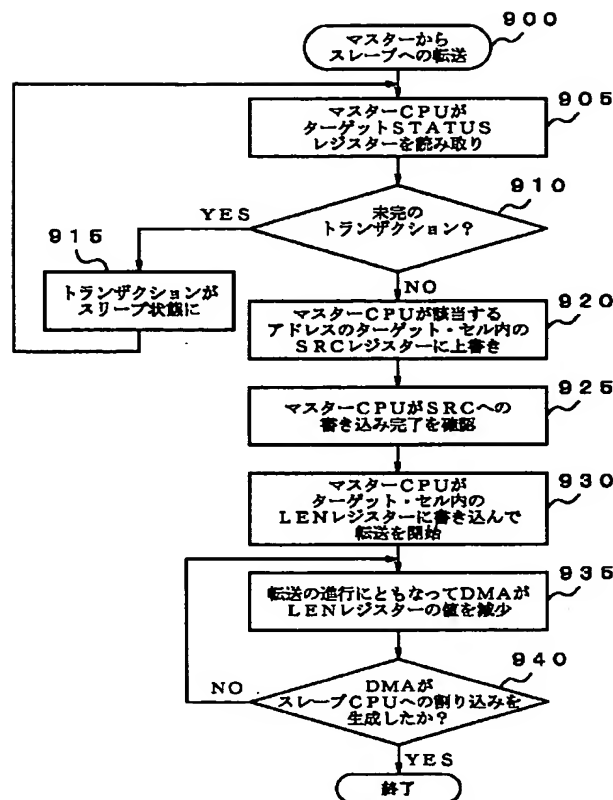
【図9】



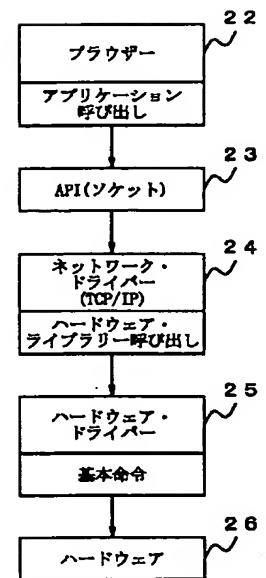
【図5】



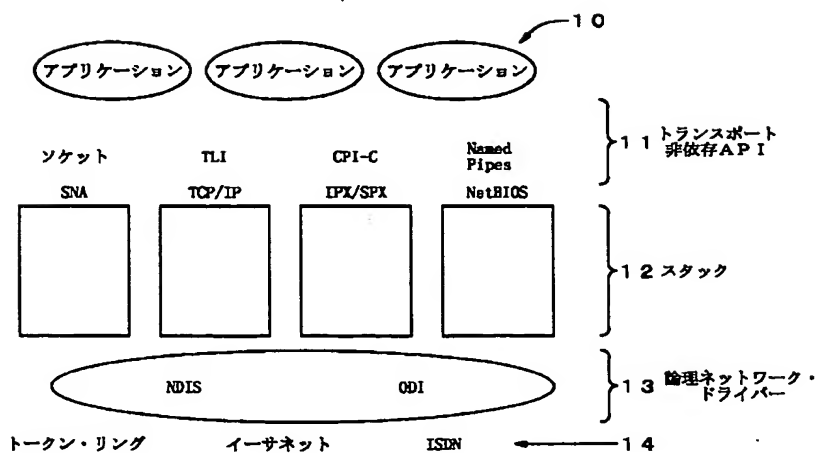
【図6】



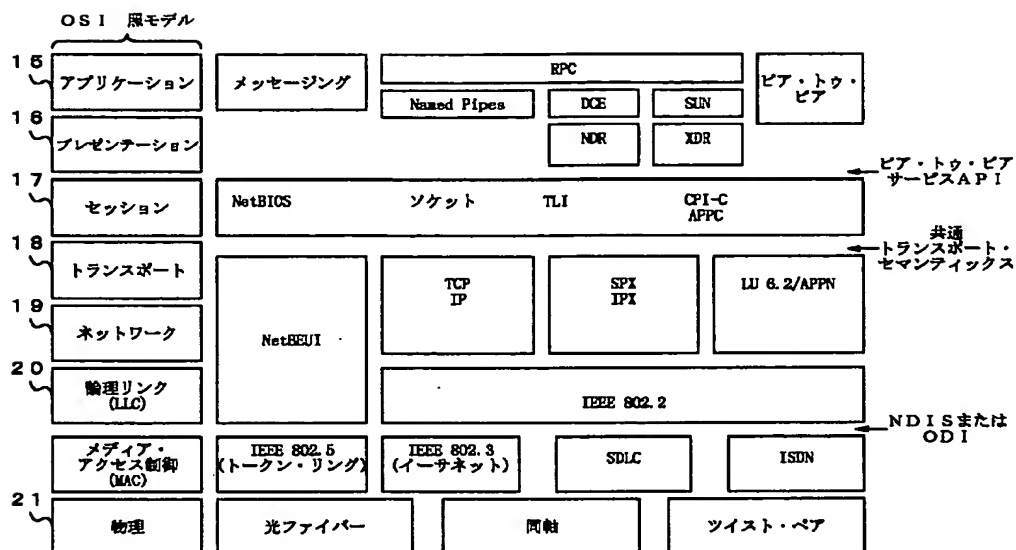
【図12】



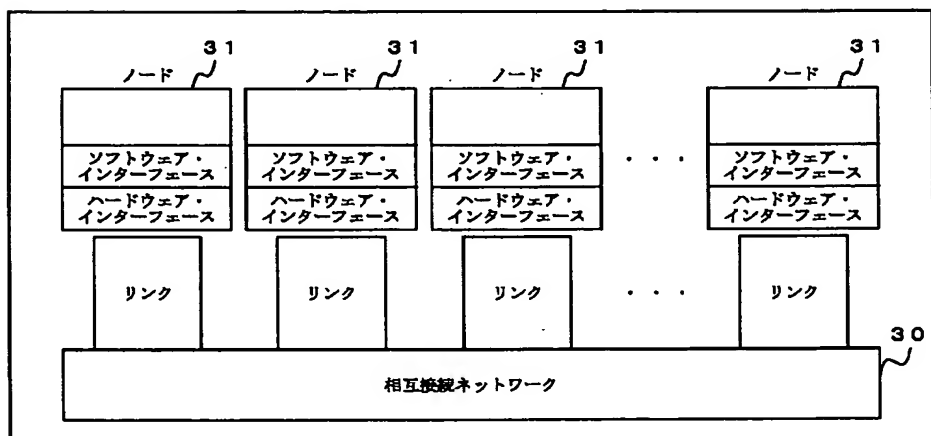
【図10】



【図 11】



【図 13】



【図 14】

